WE CLAIM:

1. A method for automatic design of a processor datapath from an input specification including a register file specification, a set of specified processor operations and a desired instruction level parallelism among the specified operations, the method comprising:

determining sets of mutually exclusive operations from the specified processor operations based on the desired instruction level parallelism;

programmatically assigning instances of functional units from a macrocell library to the sets of mutually exclusive operations, such that each specified operation is associated with a corresponding functional unit;

programmatically determining a resource allocation of register file ports to ports of the functional units; and

programmatically synthesizing register files with the allocated read/write ports and interconnects between the functional units and the allocated read/write ports.

2. The method of claim 1 wherein the ports of the functional units each have a corresponding register file port request and programmatically determining the resource allocation includes:

programmatically allocating a minimum number of read/write ports that satisfies all of the port requests.

3. The method of claim 1 wherein the specification of parallelism among operations is specified as exclusion relationships among operations that indicate which operations cannot be executed concurrently.

4. The method of claim 3 wherein the input specification further includes:

a mapping between the specified operations and register file types in the register file specification; and

operation formats describing inputs and outputs of the specified operations.

5. The method of claim 1 wherein the synthesized functional units include macrocell instances, the synthesized register files include register file instances, and the interconnect includes macrocell instances of wires, buses, muxes, or tri-states.

6. The method of claim 1 wherein determining sets of mutually exclusive operations includes:

finding maximal cliques of mutually exclusive operations based on exclusion relations derived from the input specification.

7. The method of claim 1 wherein synthesizing functional units includes:

building a list of valid functional units based on opcodes and latency of the specified operations;

from the list, selecting functional units such that each functional unit covers a maximum number of operations in a set of mutually exclusive operations.

8. The method of claim 1 including:

using the instruction level parallelism from the input specification to identify which functional unit ports can be allocated to the same

register port, and allocating selected functional unit ports to a single, shared register port.

9. A computer readable medium having software for performing the method of claim 1.

10. A method for automatic synthesis of functional units in a programmable processor datapath, the method comprising:

from an input specification defining a set of specified processor operations and instruction level parallelism among the specified operations, determining sets of mutually exclusive operations;

programmatically assigning instances of functional units from a macrocell library to the sets of mutually exclusive operations, such that each specified operation is associated with a corresponding functional unit; and

programmatically synthesizing the functional units from the macrocell library such that the functional units are described in a hardware description language.

11. The method of claim 10 wherein determining sets of mutually exclusive operations includes:

finding exclusion cliques where each clique represents a maximal set of mutually exclusive operations; and

wherein assigning instances of functional units includes programmatically selecting instances of functional units to cover the cliques from the macrocell library.

12. The method of claim 11 wherein synthesizing functional units includes:

building a list of valid functional units based on opcodes of the specified operations;

from the list, selecting functional units such that each functional unit covers a maximum number of operations in a set of mutually exclusive operations.

13. The method of claim 10 wherein functional unit instances are assigned such that the semiconductor area covered by functional units in the processor design is minimized.

14. The method of claim 10 wherein the functional unit instances are assigned such that the number of operations covered by each of the functional unit instances is maximized.

15. A computer readable medium having software for performing the method of claim 10.

16. A method for automatic synthesis of a register file and functional unit-register file interconnect in a processor, based on an input specification of register file types in the processor, specified processor operations, desired instruction level parallelism among the specified operations and functional units in the processor,

the method comprising:

for each type of register file specified in the processor, establishing a set of read/write port requests between the functional units and each of the register file types;

programmatically computing a resource allocation of register ports in the register file types to read/write port requests, including

determining how to share a register port for two or more functional unit ports based on the specification of instruction level parallelism among the operations; and

programmatically synthesizing register files with the allocated read/write ports and interconnects between the functional units and the allocated read/write ports.

17. The method of claim 16 wherein the resource allocation uses a contiguous allocation heuristic that simplifies interconnect layout by allocating register port requests from a functional unit to contiguous register ports.

18. A computer readable medium having software for performing the method of claim 16.